

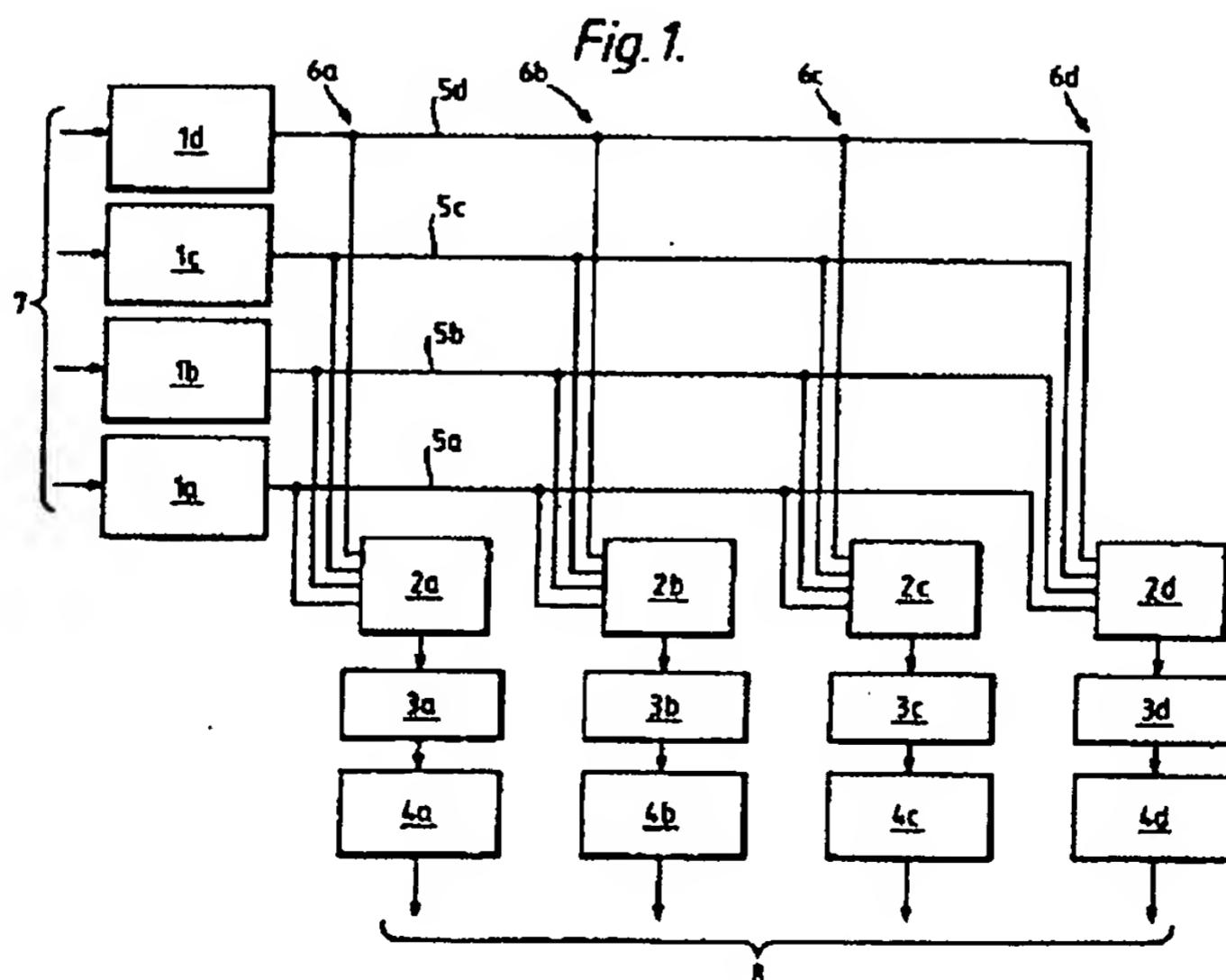
(12) UK Patent Application (19) GB (11) 2 237 908 (18) A

(43) Date of A publication 15.05.1991

(21) Application No 9020776.2	(51) INT CL <sup>8</sup> G06F 15/00
(22) Date of filing 24.09.1990	(52) UK CL (Edition K) G4A ACF AMP
(30) Priority data (31) 8925227 (32) 08.11.1989 (33) GB	(56) Documents cited None
(71) Applicant British Aerospace Public Limited Company (Incorporated in the United Kingdom) 11 Strand, London, WC2N 5JT, United Kingdom	(58) Field of search UK CL (Edition K) G4A ACF ACL AMP AMV, MAT TCGG TCGX TCJA TCXX INT CL <sup>8</sup> G06F
(72) Inventor Steven Maxwell Parkes	
(74) Agent and/or Address for Service P B Rooney Corporate Intellectual Property Department, British Aerospace Plc, PO Box 87, Royal Aerospace Est, Farnborough, GU14 6YU, United Kingdom	

(54) Parallel processing of data

(57) In parallel processing of data, the data is organised into a two dimensional array having at least two rows (5a, 5b, 5c, 5d) and at least two transverse linking columns (6a, 6b, 6c, 6d), first high level data processing is carried out by first processing means on the rows or on the columns, corner turning is carried out on the first processed data to turn it from said rows into said columns or vice versa, and second high level data processing is carried out by second processing means on the corner turned data in said columns or in said rows, with the first processed data in said rows or columns being stored, before or after corner turning, in separate memories (3a, 3b, 3c, 3d) associated one with each row (5a, 5b, 5c, 5d) or column (6a, 6b, 6c, 6d).



This print incorporates corrections made under Section 117(1) of the Patents Act 1977.

GB 2 237 908 A

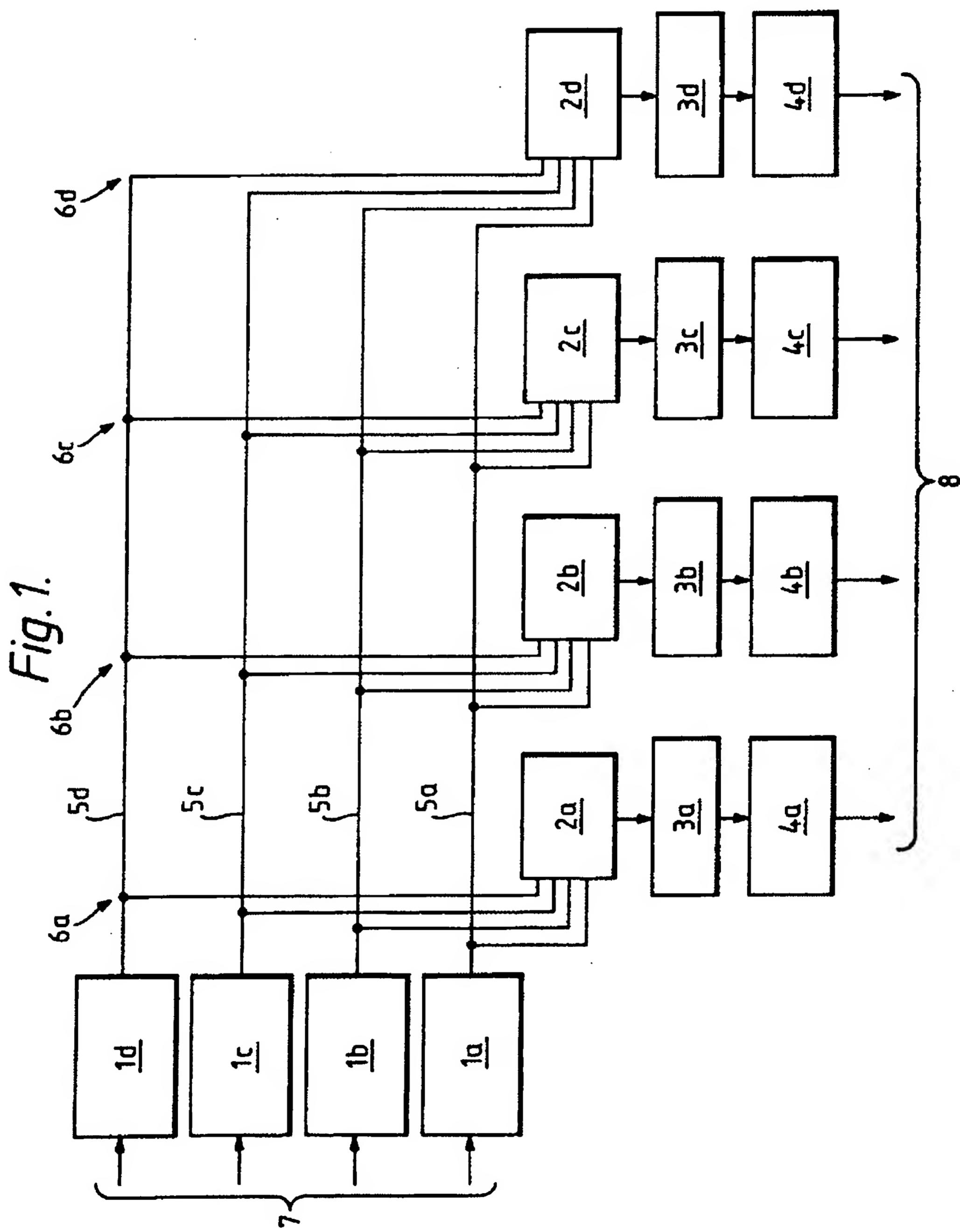
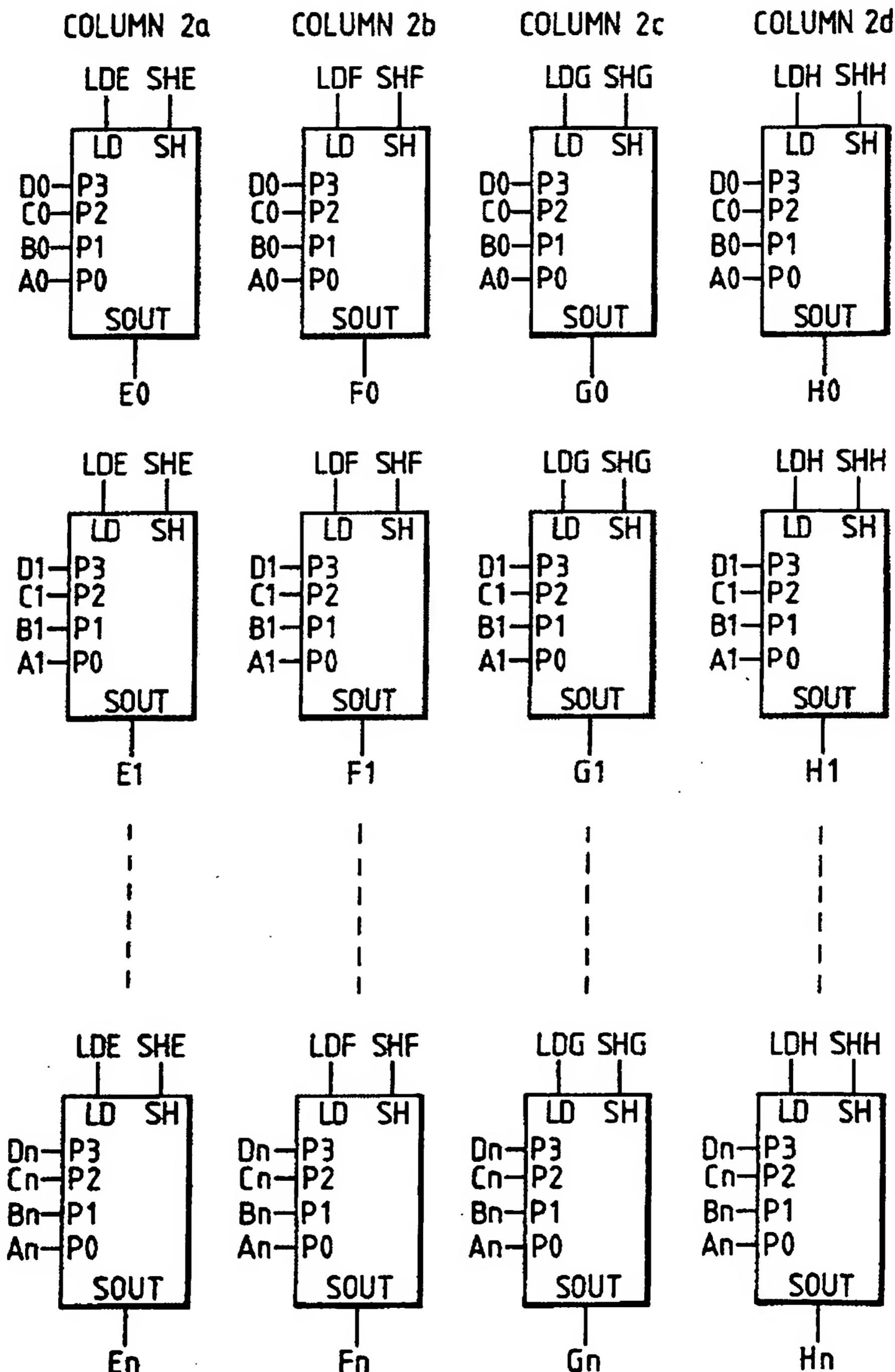


Fig. 2.



3/4

*Fig. 3.*

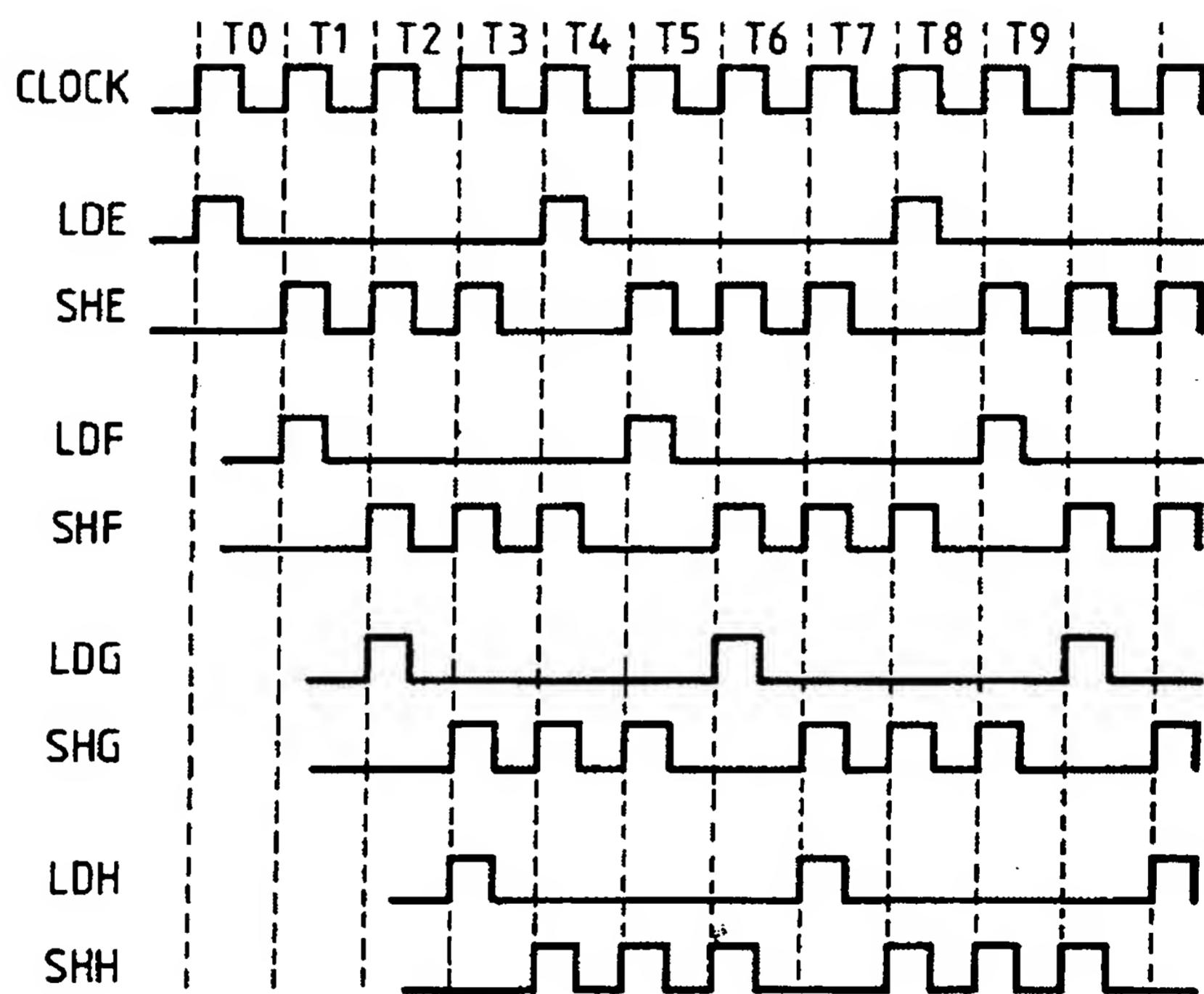
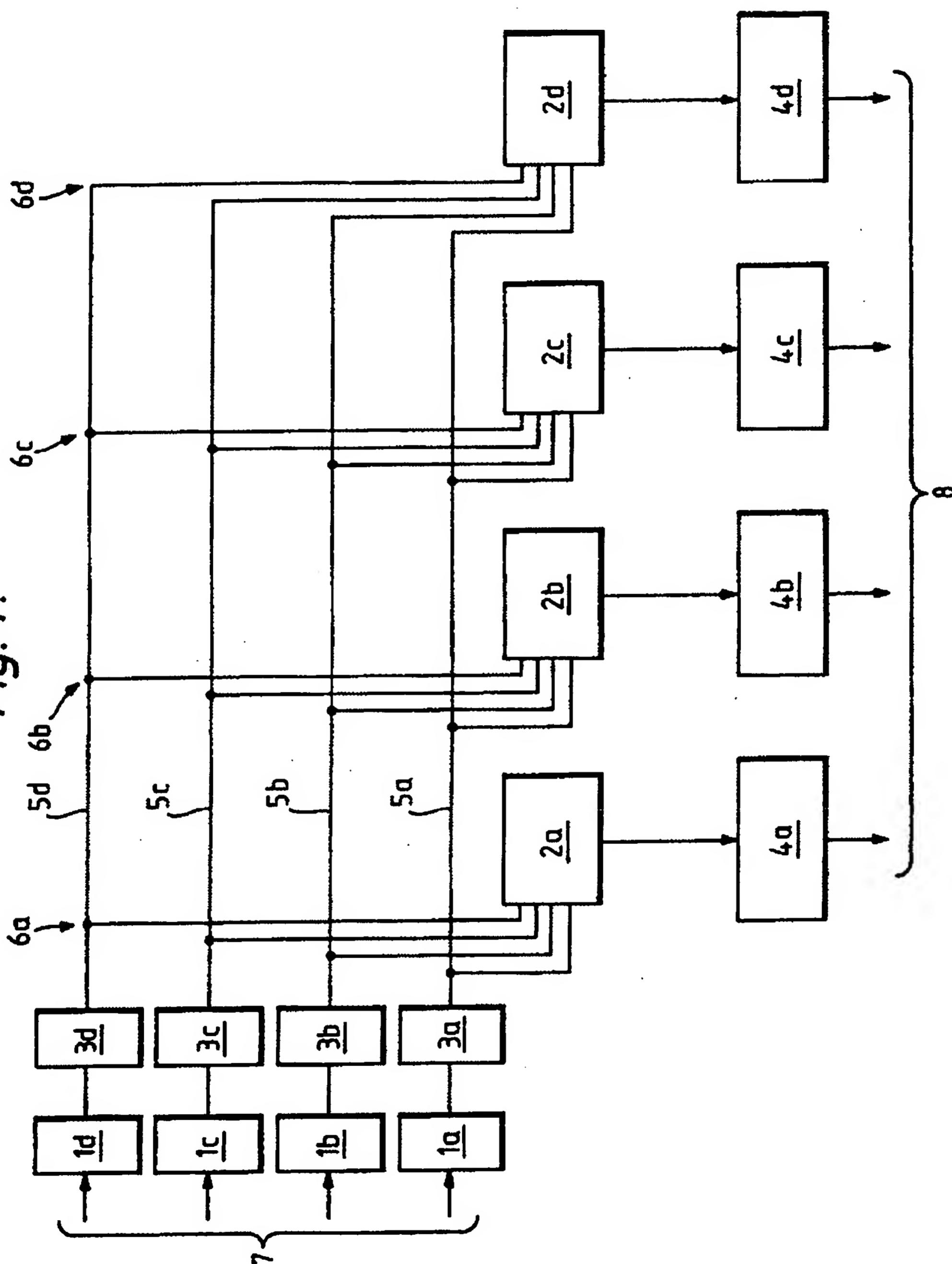


Fig. 4.



Method and Apparatus for Parallel Processing Data

This invention relates to a Method and Apparatus for parallel processing data, particularly, but not exclusively, suitable for the processing of signal and/or image data.

Data is commonly stored serially row by row on a direct access bulk storage peripheral such as a disc file unit. Such data may be transferred to or from the disc file in blocks which are stored at random on the disc. Thus if it is required to access the columns of a matrix stored row by row, many blocks will require retrieval from the disc to access the column elements. This is time consuming and inefficient.

One way of reorganising the stored data is to transpose the data so that the stored blocks contain data in serial column order instead of serial row order. This reorganisation is termed 'corner turning'. Conventionally such corner turning has been implemented by writing the row ordered data into a single large memory and then reading it out in column order using a "column ordered" address generator. However this known technique has the disadvantage of causing a communications bottleneck.

There is thus a need for a generally improved method and apparatus for parallel processing of data which is more efficient and which causes less of a communications bottleneck than the aforementioned conventional techniques.

According to one aspect of the present invention there is provided a method of parallel processing data, in which the data is organised into a two dimensional array having at least two rows and at least two transverse linking columns, first high

level data processing is carried out on the rows or on the columns, corner turning is carried out on the first processed data to turn it from said rows into said columns or vice versa, and second high level data processing is carried out on the corner turned data in said columns or in said rows, with the first processed data in said rows or columns being stored, before or after corner turning, in separate memories associated one with each row or column.

Thus the corner turning memory is distributed between two or more column processing elements. By operating all the memories in parallel the communications bottleneck caused by a single large corner turning memory is overcome.

Preferably said first high level data processing is carried out on each of said rows of data, the corner turning is carried out on the processed row data to turn it into column ordered data and said second high level data processing is carried out on the column ordered data.

Conveniently said first high level processing is carried out by one row processor per row, said second high level processing is carried out by one column processor per column and the processed row data is stored, in said separate memories associated one with each row, before corner turning.

Advantageously said first high level processing is carried out by one row processor per row, said second high level processing is carried out by one column processor per column and the processed row data is stored in separate memories associated one with each column after corner turning.

Preferably corner turning is carried out by feeding the processed data from each row in sequence, in parallel into a shift register associated one with each column to form a series of data sets and shifting the series of data sets from each shift register into the associated memory in column order, from whence the column ordered data can be read by the associated column processor.

Conveniently said first high level processing is carried out on each of said columns of data, the corner turning is carried out on the processed column data to turn it into row ordered data and said second high level processing is carried out on the row ordered data.

Advantageously said first high level processing is carried out by one column processor per column, said second high level processing is carried out by one row processor per row and the processed column data is stored after corner turning in said separate memories associated one with each row.

Preferably the corner turning is carried out by feeding the processed data from each column in sequence, in parallel into a shift register associated one with each row to form a series of data sets and shifting the series of data sets from each shift register into the associated memory in row order, from whence the row ordered data can be read by the associated row processor.

Conveniently one dimensional Fast Fourier Transforms are carried out on the data in each processor.

According to a second aspect of the present invention there is provided apparatus for the parallel processing of data, including means for organising data into a two dimensional array having at least two rows and at least two transverse linking columns, first processing means for carrying out first high level data processing on the rows or the columns, corner turning means for carrying out corner turning on the first processed data to turn it from said rows into said columns or vice versa, second processing means for carrying out second high level data processing on the corner turned data in said columns or in said rows, and at least two separate memories associated one with each row or column, which memories are located and operable to store the first processed data in said rows or columns before or after corner turning.

Preferably the first and second processing means are data processors located one in each row and column and wherein the corner turning means includes a plurality of shift registers located one in each column.

Conveniently the array has at least two substantially parallel rows, with the first processing means data processors being located respectively one at each input end of each row, with the output end of each row being connected to the shift register of one column and with the rows being connected intermediate the row ends to the shift register of another column, and wherein the second processing means data processors are located respectively one at each output end of each column to receive the output from the associated shift

register.

Advantageously the memories are located one in each row between the associated row data processor and the row connections to the column shift register most remote from the output ends of the rows.

Preferably the memories are located one in each column between the associated column data processor input and the associated column shift register output.

Conveniently each data processor is operable to carry out one dimensional Fast Fourier Transforms.

For a better understanding of the present invention, and to show how the same may be carried into effect, reference will now be made, by way of example, to the accompanying drawings, in which:

Figure 1 is a block diagram of apparatus according to a first embodiment of the invention for parallel processing data,

Figure 2 is a diagram illustrating an arrangement of shift registers to achieve corner turning of data using the method according to the present invention and the apparatus of Figure 1,

Figure 3 is a diagram illustrating the relative timing of the control signals used by the shift register arrangement of Figure 2,

Figure 4 is a view similar to that of Figure 1 showing a block diagram of an apparatus for parallel processing data according to a second embodiment of the invention.

As shown in the accompanying drawings, the apparatus and method of the invention for parallel processing of data, such as signal and/or image data, basically involves organising the data into a two dimensional array having at least two rows and at least two transverse linking columns. In the embodiment illustrated in Figures 1 and 4 there are four such rows 5a, 5b, 5c and 5d and four such columns 6a, 6b, 6c, 6d. First high level data processing is carried out on the rows, 5a, 5b, 5c, 5d or on the columns 6a, 6b, 6c, 6d, corner turning is carried out on the first processed data to turn it from the rows into the columns or vice versa and second high level data processing is carried out on the corner turned data in the columns or in the rows. The first processed data in the rows 5a, 5b, 5c, 5d or in the columns 6a, 6b, 6c, 6d is stored before or after corner turning, in separate memories 3a, 3b, 3c, 3d associated one with each row or column.

In the embodiment illustrated in Figure 1 the first high level data processing is carried out on each of the rows 5a, 5b, 5c, 5d by one row processor 1a, 1b, 1c, 1d and the second high level processing is carried out on the column ordered data by one column processor 4a, 4b, 4c, 4d. The corner turning is carried out on the processed row data by a plurality of shift registers 2a, 2b, 2c, 2d located respectively one in each column 6a, 6b, 6c, 6d. The processed row data is stored in separate memories 3a, 3b, 3c, 3d associated one with each column, after corner turning. Although in the illustrated embodiments of

Figures 1 and 4 four rows and four columns have been shown, it is of course to be understood that the method and apparatus of the invention is operable with at least two rows and at least two columns.

Corner turning is carried out by feeding the processed data from each row 5a, 5b, 5c, 5d in sequence, in parallel into the associated shift register associated one with each column to form a series of data sets. The series of data sets for each shift register 2a, 2b, 2c, 2d is shifted into the associated memory 3a, 3b, 3c, 3d in column order, from whence the column ordered data can be read by the associated column processor 4a, 4b, 4c, or 4d.

The row and column processors have a high functionality, performing complete operations on segments of data (for example 256 samples) rather than elementary operations on single data samples. In the Figure 1 embodiment each column processor 4a, 4b, 4c, 4d has an associated memory 3a, 3b, 3c, 3d into which the corner turned data is stored prior to column processing. Thus the total memory required to hold the data is distributed between all the column processors.

This provides a high bandwidth communications structure connecting a parallel array of row processors with a concurrently operating array of column processors. Extremely high performance may be obtained without communication bottlenecks, with the addition of further rows and columns and thus further row processors and column processors, automatically increasing

the data input/output bandwidth. One dimensional data may be processed by organising it in a two dimensional form prior to processing. Data in three or more dimensions may be processed by first organising the data into two dimensional arrays of data.

Although not illustrated, the first high level processing could be carried out on each of the columns of data, the corner turning carried out on the processed column data to turn it into row ordered data and the second high level processing carried out on the row ordered data. In other words the sequence of Figure 1 in which data is inputted at 7 and outputted at 8 could be reversed. In such an alternative, the first high level processing would be carried out by the column processors, the second high level processing carried out by the row processors and the processed column data stored, after corner turning, in the separate memories associated one with each row. The Figure 4 embodiment illustrates such alternative apparatus in which the memories are associated with the row processors although in the illustrated Figure 4 embodiment the data input 7 is to the rows and the data output 8 is from the columns.

#### Example 1

The example algorithm used in the method of the invention is the two dimensional Fast Fourier Transform (FFT). This is a well known algorithm which may be implemented by first applying a one dimensional FFT to all the rows (5a, 5b, 5c, 5d) of the two dimensional data array followed by applying a one dimensional FFT to all the columns (6a, 6b, 6c, 6d) of the resultant data array. In this example a 64 by 64 point array of

data as shown in Table 1 is to be transformed by processor apparatus according to the first embodiment of the invention as illustrated in Figure 1.

In this particular case the row processors 1a, 1b, 1c, 1d and the column processors 4a, 4b, 4c, 4d all perform identical functions which is a 64 point one dimension FFT.

TABLE 1.

0.0	0.1	0.2	0.3	0.4	0.5	...	..
1.0	1.1	1.2	1.3	1.4	1.5	...	..
2.0	2.1	2.2	2.3	2.4	2.5	...	..
3.0	3.1	3.2	3.3	3.4	3.5	...	..
4.0	4.1	4.2	4.3	4.4	4.5	...	..
5.0	5.1	5.2	...	...	...	...	..
6.0	6.1	...	...	...	...	...	..
7.0	...	...	...	...	...	...	..
8.0	...	...	...	...	...	...	..
...	...	...	...	...	...	...	..
60.0	...	...	...	...	...	...	..
61.0	61.1	...	...	...	...	...	..
62.0	62.1	62.2	...	...	...	...	..
63.0	63.1	63.2	63.3	63.4	63.5	...	..

The data was processed four rows at a time. The first four rows of data were passed through the four row processors 1a, 1b, 1c, 1d which perform 64 point FFTs on the data rows 5a, 5b, 5c, 5d respectively. The row processors output their results in the same order as the data went in. The first set of data to emerge was  $\{0,0\}$  from row processor 1a,  $\{1,0\}$  from row processor 1b,  $\{2,0\}$  from row processor 1c and  $\{3,0\}$  from row processor 1d. This set of data was loaded in parallel into the first shift register 2a, then shifted out and placed in memory 3a. The next set of data from the row processors  $\{\{0,1\} \{1,1\} \{2,1\} \{3,1\}\}$  was loaded onto the next shift register 2b, then shifted out into memory 3b. In a similar way memory 3c will receive the data  $\{\{0,2\} \{1,2\} \{2,2\} \{3,2\}\}$  and memory 3d will receive data  $\{\{0,3\} \{1,3\} \{2,3\} \{3,3\}\}$ . The next set of data to emerge from the row processors,  $\{\{0,4\} \{1,4\} \{2,4\} \{3,4\}\}$  was loaded by the first shift register 2a into memory 3a. After rows 5a to 5d had been processed the next four rows were processed starting at 4,0 then 5,0 then 6,0 and 7,0. This procedure continued with the row processors processing each set of four rows of data in turn until the last set of data  $\{\{60,63\} \{61,63\} \{62,63\} \{63,63\}\}$  had been loaded into memory 3d.

Now memory 3a contains all the data from every fourth column of the data array starting at column 6a (i.e. columns 1, 5, 9....) and memories 3c and 3d contain all the data from every fourth column starting at columns 6c and 6d respectively.

TABLE 2

0,0	0,1	0,2	0,3	0,4	0,5	...	..
1,0	1,1	1,2	1,3	1,4	1,5	...	..
2,0	2,1	2,2	2,3	2,4	2,5	...	..
3,0	3,1	3,2	3,3	3,4	3,5	...	..
4,0	4,1	4,2	4,3	4,4	4,5	...	..
5,0	5,1	5,2	...	...	...	...	..
6,0	6,1	...	...	...	...	...	..
7,0	...	...	...	...	...	...	..
8,0	...	...	...	...	...	...	..
...	...	...	...	...	...	...	..
60,0	...	...	...	...	...	...	..
61,0	61,1	...	...	...	...	...	..
62,0	62,1	62,2	...	...	...	...	..
63,0	63,1	63,2	63,3	63,4	63,5	...	..

The column processors 4a, 4b, 4c, and 4d can now read the column orientated data out of the memories 3a, 3b, 3c and 3d respectively and process each column in turn. The column processors 4a, 4b, 4c and 4d first process columns 6a, 6b, 6c, 6d (0, 1, 2 and 3) respectively, followed by successive columns (4, 5, 6 and 7) and so on until all the columns of data have been processed. The column processors will perform 64 point FFTs on each column of data in the example two dimensional FFT algorithm. The data from the processing apparatus appears in column order at the output of the column processors. If desired a further parallel processing apparatus may be added to the output of the column processors 4a, 4b, 4c, 4d to convert the column ordered data back to row ordered form.

By using a shift register structure to perform the corner turning the memory elements required to hold the corner turned data before column processing are distributed evenly between the four column processors 4a, 4b, 4c, 4d. The four memories 3a, 3b, 3c, 3d are accessed concurrently, thereby improving data throughput compared with a conventional single memory arrangement.

Further to illustrate the method of the present invention a specific implementation of the shift register structure will now be described. With reference to Figure 2 an array of four-bit shift registers were connected to the outputs of the row processors 1a, 1b, 1c, 1d and to the inputs of the column memories 3a, 3b, 3c, 3d. The data output from the row processor 1a is represented by A0, a1, A2, ... where A0 is the least significant

bit of the data word, A1 the next bit and so on. Similarly B0, B1, B2,... is the output from row processor 1b; C0, C1... the output from row processor 1c and D0, D1... the output from row processor 1d.

The input to column memory 3a is represented by E0, E1, E2,... where E0 is the least significant bit, E1 the next bit and so on. F0, F1,...; G0, G1,... and H0, H1,... are the inputs to column memories 3b, 3c and 3d respectively.

Each four bit shift register is controlled by two signals, LD and SH. LD causes the data at the parallel input (P0, P1, P2, P3) of the shift register to be parallel loaded into the register. SH causes the data within the shift register to be shifted down one position. The serial (shifted) data appears at the output, SOUT, of the shift register. Each vertical bank of shift registers in Figure 2 have common LD and SH control signals. For example the first bank (column 2a) which generates the corner turned signals E0, E1,... for column memory 3a uses the signals SHE and LDE. The relative timing of the shift register control signals is shown in Figure 3.

The operation of the shift register "corner turning" structure will now be described. As the first set of data emerges from the row processors 1a, 1b, 1c, 1d during clock period T0 (see Figure 3) the LDE (load shift register 2a) signal is activated. This causes the data from all the row processors to be loaded into the first column of shift registers (column 2a in Figure 2). If a 16 bit word is used as the output from each row processor then there will be sixteen shift registers in the

column, i.e. one shift register for each bit. Since there are four row processors 1a, 1b, 1c, 1d in this example each shift register 2a, 2b, 2c, 2d will be four bits long. Once the data has been loaded into the first column of shift registers (column 2a) the data from row processor 1a (A0, A1,...) is immediately available at the outputs of those shift registers (E0, E1,...).

During the next clock period T1 the next set of data emerges from the row processors and is loaded into the second column of shift registers (column 2b) by signal LDF. At the same time the SHE line is activated shifting the data in the first column of shift registers (column 2a) down one place so that the data previously loaded from row processor 1b is available at their outputs.

On the next clock pulse (T2) the third set of data from the row processor is loaded into the third column of shift registers (column 2c) by signal LDG. Signals SHE and SHF cause the data in the first and second column of shift registers (columns 2a and 2b) respectively) to be shifted down one place.

Now data C0, C1,... loaded in time slot T0 is available at the output of the first column of shift registers (column 2a), data B0, B1,... (loaded in time slot T2) is available at the output of the third column of shift registers (column 2c).

This procedure continues with data from the row processors being loaded into one column of shift registers while the data in the other three columns of shift registers is shifted down one place. The output from the shift registers constitutes the required corner turned data for each column processor 4a, 4b,

4c, 4d which is loaded into its associated memory 3a, 3b, 3c, 3d.

Although the foregoing Example 1 has been described in terms of the apparatus for parallel processing of data according to the embodiment of Figure 1, it is to be understood that a similar method can be carried out with the apparatus for the parallel processing of data as illustrated in the second embodiment of Figure 4. The primary difference between the two embodiments is that in the embodiment of Figure 4 the memories 3a, 3b, 3c and 3d are associated with the row processors 1a, 1b, 1c and 1d. Additionally although in the two illustrated embodiments the data input has been shown as to the row processors 1a, 1b, 1c and 1d, with the output from the column processors 4a, 4b, 4c and 4d, it is, however, to be understood that the data input could be to the column processors and the data output from the row processors.

Additionally, although four rows 5a, 5b, 5c and 5d and four columns 6a, 6b, 6c and 6d have been described and illustrated with respect to the embodiments of Figures 1 and 4 a minimum of two such rows and two such columns may be provided or more than four such rows and columns if desired. In any event each row will include one row processor and each column will include one shift register and column processor. One memory will be provided for each column or row. The output ends of the rows 5a, 5b, 5c and 5d are connected, in the illustrated embodiments, to the shift register 2d of the column 6d. The rows are also connected intermediate the row ends at

specific spacings there along to the shift register 2a of the column 6a, to the shift register 2b of the column 6b and to the shift register 2c of the column 6c. In the Figure 1 embodiment the memories 3a, 3b, 3c and 3d are connected respectively one between each of the shift registers and the column processors. In the Figure 4 embodiment the memories are connected one between each of the row processors and the first column connection 6a. Each row processor and each column processor is operable to carry out one dimensional Fast Fourier Transforms.

The column processors or row processors each or all may be capable of performing one specific function only, which preferably may be selected from several possible predefined modes of operation.

CLAIMS

1. A method of parallel processing data, in which the data is organised into a two dimensional array having at least two rows and at least two transverse linking columns, first high level data processing is carried out on the rows or on the columns, corner turning is carried out on the first processed data to turn it from said rows into said columns or vice versa, and second high level data processing is carried out on the corner turned data in said columns or in said rows, with the first processed data in said rows or columns being stored, before or after corner turning, in separate memories associated one with each row or column.
2. A method according to Claim 1, in which said first high level data processing is carried out on each of said rows of data, the corner turning is carried out on the processed row data to turn it into column ordered data and said second high level data processing is carried out on the column ordered data.
3. A method according to Claim 2, in which said first high level processing is carried out by one row processor per row, said second high level processing is carried out by one column processor per column and the processed row data is stored in said separate memories associated one with each row, before corner turning.
4. A method according to Claim 2, in which said first high level processing is carried out by one row processor per row, said second high level processing is carried out by one column processor per column and the processed row data is stored in

separate memories associated one with each column, after corner turning.

5. A method according to Claim 4, in which corner turning is carried out by feeding the processed data from each row in sequence, in parallel into a shift register associated one with each column to form a series of data sets and shifting the series of data sets from each shift register into the associated memory in column order, from whence the column ordered data can be read by the associated column processor.

6. A method according to Claim 1, in which said first high level processing is carried out on each of said columns of data, the corner turning is carried out on the processed column data to turn it into row ordered data and said second high level processing is carried out on the row ordered data.

7. A method according to Claim 6, in which said first high level processing is carried out by one row processor per row and the processed column data is stored after corner turning in said separate memories associated one with each row.

8. A method according to Claim 7, in which the corner turning is carried out by feeding the processed data from each column in sequence, in parallel into a shift register associated one with each row to form a series of data sets from each shift register into the associated memory in row order, from whence the row ordered data can be read by the associated row processor.

9. A method according to Claim 3, Claim 4 or Claim 7, in which one dimensional Fast Fourier Transforms are carried out on the data in each processor.

10. A method according to any one of Claims 1 to 9, in which one dimensional data is processed by first organising it into a two dimensional array.
11. A method according to any one of Claims 1 to 10, in which the data to be parallel processed is signal and/or image data.
12. A method according to any one of Claims 1 to 9, in which data in three or more dimensions is processed by first organising it into two dimensional arrays of data.
13. A method of parallel processing data substantially as hereinbefore described with reference to Figures 1 to 3 or Figure 4 of the accompanying drawings.
14. Apparatus for the parallel processing of data, including means for organising data into a two dimensional array having at least two rows and at least two transverse linking columns, first processing means for carrying out first high level data processing on the rows or the columns, corner turning means for carrying out corner turning on the first processed data to turn it from said rows into said columns or vice versa, second processing means for carrying out second high level processing on the corner turned data in said columns or in said rows, and at least two separate memories associated one with each row or column, which memories are located and operable to store the first processed data in said rows or columns before or after corner turning.
15. Apparatus according to Claim 14, wherein the first and second processing means are data processors located one in each

row and column and wherein the corner turning means includes a plurality of shift registers located one in each column.

16. Apparatus according to Claim 15, wherein the array has at least two substantially parallel rows, with the first processing means data processors being located respectively one at each input end of each row, with the output end of each row being connected to the shift register of one column and with the rows being connected intermediate the row ends to the shift register of another column, and wherein the second processing means data processors are located respectively one at each output end of each column to receive the output from the associated shift register.

17. Apparatus according to Claim 16, wherein the memories are located one in each row between the associated row data processor and the row connections to the column shift register most remote from the output ends of the rows.

18. Apparatus according to Claim 16, wherein the memories are located one in each column between the associated column data processor input and the associated column shift register output.

19. Apparatus according to any one of Claims 15 to 18, wherein each data processor is operable to carry out one dimensional Fast Fourier Transforms.

20. Apparatus for the parallel processing of data, substantially as hereinbefore described and as illustrated in Figure 1 or Figure 4 of the accompanying drawings.